

Praxisorientierte Einführung in C++

Lektion: "Allgemeines"

Christof Elbrechter

Neuroinformatics Group, CITEC

April 12, 2012

Table of Contents

- Geschichtliches
- Einordnung
- C++: Pro/Kontra

Geschichtliches

- ▶ Nachfolger von C und Objective-C
- ▶ Name “C++” ist Wortspiel, welches “C” beinhaltet und den Post-Inkrement-Operator “++”
- ▶ Entwickelt seit 1979
 - Von Bjarne Stroustrup bei “AT&T”
- ▶ Originalname “C with Classes”

Geschichtliches cntd.

- ▶ Ursprüngliche Motivation war Forschung im Bereich Betriebssysteme und verteiltes Rechnen
- ▶ 1983 Umbenennung in “C++”
- ▶ 1985: Version 1, 1989: Version 2
- ▶ 1990 Erscheinen des Buches “The Annotated C++ Reference Manual” auf dessen Basis die spätere Standardisierung geschah

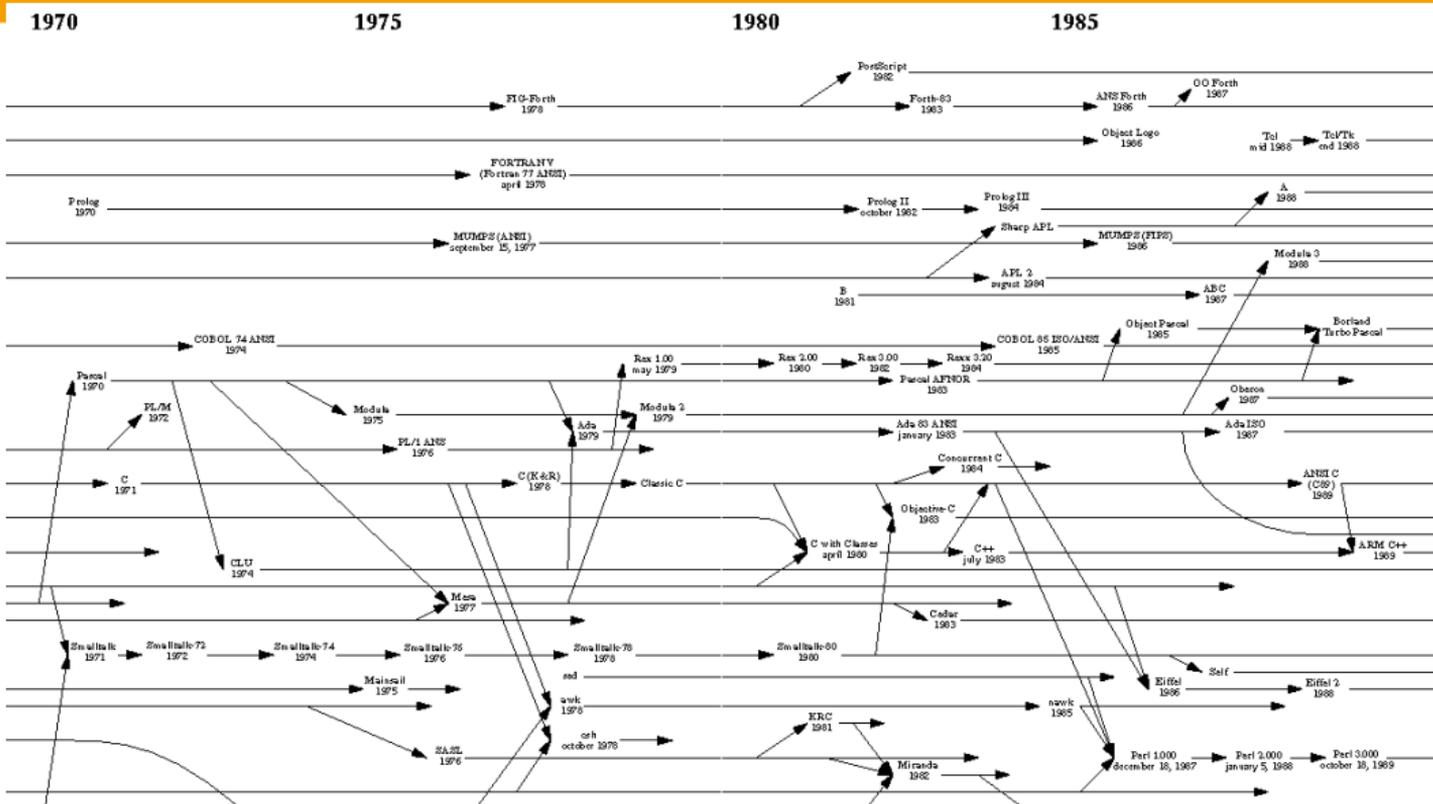
Geschichtliches cntd.

- ▶ 1998 Standardisierung im ISO-Standard: "ISO/IEC 14882:1998"
- ▶ 2003 Überarbeitung des Standards: "ISO/IEC 14882:2003"
- ▶ 2011 C++11 Standard
(früher als C++0x gehandelt)
 - Closures
 - Lambda Expressions
 - Move Semantics
 - rvalue References
 - implicit type-binding (Keyword auto)
 - uvm. (hier wird es einen eigenen Vorlesungstermin für geben)

Einordnung

- ▶ Kompilierte Sprache (z.B. Gcc, MS-Visual-C++, MingW,etc)
- ▶ Statisch typisiert (und explizit, in C++11 auch implizit)
- ▶ Multiparadigmensprache
 - Strukturiert
 - Objektorientiert
 - Imperativ
 - Generisch
 - Ein kleines bisschen funktional (in C++11 etwas funktionaler)
- ▶ Mächtige High-Level-Konstrukte kombiniert mit maschinennahen Operationen

Einordnung



C++: Pro/Kontra

- ▶ Performanz (in Spezialfällen fast 100 mal schneller als interpretierte Sprachen wie z.B. Java)
- ▶ Sehr portabel (durch maschinenabhängige Datentypen)
 - Auf vielen Architekturen implementiert
 - Nachteil: Semantik kann variieren
- ▶ General Purpose

C++: Pro/Kontra

- ▶ Deutlich bessere Strukturierungs- und Abstraktionsmöglichkeiten als C
 - Objekt-Orientierung
 - Vererbung (auch mehrfach)
 - Templates
 - (Operator)-Überladung
- ▶ (Fast) Obermenge von C
 - Leichter Einstieg für C-Programmierer
 - (Bjarne Stroustrup: Bringt wenig!)

C++: Pro/Kontra

- ▶ Viele Bibliotheken verfügbar (insbesondere auch (fast) alle C-Bibliotheken)
- ▶ Immer noch sehr verbreitet (insb. in der Forschung und Spiele-Industrie)
- ▶ Sehr maschinennah
- ▶ C gilt auch als “plattformunabhängiger Assembler”
- ▶ Erlaubt sogar Maschinencode durch das Keyword `asm` einzubinden
 - Nachteil: Implementationsabhängig, nicht portabel

C++: Pro/Kontra

- ▶ Kein implizites/automatisches Speichermanagement (Garbage Collector)
- ▶ Relativ schlechte Unterstützung für funktionale Programmierung:
 - Keine Closures, Lambda-Ausdrücke
 - Aber Typpolymorphismus durch Vererbung und Templates
- ▶ Mangelhafte Introspektion/Reflexion
- ▶ Vergleichsweise laaaange Übersetzungszeit (verglichen mit z.B. C und Java)

C++: Pro/Kontra

- ▶ Nicht verzweifeln: C++ ist sehr komplex (>50 Schlüsselwörter¹)
 - Man braucht nicht alles können
 - Es gibt nur wenige richtige Experten
 - Der C++-Standard ist 800 Seiten stark
 - An vieles muss man sich gewöhnen
- ▶ Auch ich werde hier Fehler machen (absichtlich natürlich :-)
- ▶ Teilweise wird 100%ige Korrektheit gegen Verständlichkeit eingetauscht.
- ▶ Referenz ist und bleibt der C++-Standard

¹und einige werden dazu noch mehrfach verwendet